UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/717,676	11/21/2000	Bradley J. Bartz	777.346US1	9183

7590 07/17/2003

Steven J Rocci
Woodcock Washburn Kurtz Mackiewicz & Norris LLP
One Liberty Place
46th Floor
Philadelphia, PA 19103

EXAMINER	
VU, TUAN A	
ART UNIT	PAPER NUMBER
2124	6

DATE MAILED: 07/17/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/717,676	BARTZ ET AL.	
	Examiner	Art Unit	
	Tuan A Vu	2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *21 November 2000* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-32* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-32* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *21 November 2000* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) 5 .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

1. This action is responsive to the application filed November 21, 2000.

Claims 1-32 have been submitted for examination.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 3-12, and 15 are rejected under 35 U.S.C. 102(b) as being anticipated by Skinner, USPN: 5,481,722 (hereinafter Skinner).

As per claim 3, Skinner discloses a method useful in developing software having multiple versioned documents, comprising:

defining at least 2 nested types of subdivision in both documents (e.g. *start control line*, *end control line* – Fig. 6; *text line* – Fig. 6 – Note: second subdivision could be line-by-line or character-by-character in a line);

comparing current first-type subdivision of one documents to a counterpart of other document; and indicating differences therebetween (e.g. *output table* , *branch deltas* – Fig 7a-b);

comparing current second-type subdivision of one documents to a counterpart of the other document (Fig. 6 – Note: the comparing of lines by lines text is implicitly disclosed);

repeating second comparing step within subdivision of first type (e.g. *text lines* - Fig. 6);

repeating first comparing step for further first-type subdivision within the documents (Fig. 6; *control line* - col. 7, line 30 to col. 8, line 25);

producing an output document indicating differences found in both comparing steps (e.g. *output delta table* - col. 8, lines 16-25; Fig. 5)

As per claim 4, as suggested by Skinner, each control line is indicative of a revision/edition level so such control line entails a matched version (e.g. *unchanged, may be zero* -- col. 7, lines 37-53). Skinner has implicitly disclosed that upon detecting no difference between first-type subdivision, the comparing of second-type subdivision is inhibited.

As per claims 5 and 6, Skinner discloses that the first subdivision type is a line(*start control line*, *end control line* – Fig. 6; *text line* – Fig. 6). But Skinner does not explicitly teach that the second subdivision is a character but since a character matching is inherent to a line comparison, Skinner has implicitly disclosed that the second subdivision type is a line.

As per claim 7, this is a computer-readable medium version of claim 3. As for a medium to support the software product, Skinner (Fig. 3) discloses use of workstations to develop the software building, hence implicitly discloses the use of computer-readable medium.

As per claim 8, Skinner discloses a method useful in developing software having multiple versioned documents, comprising:

defining at least 3 nested types of subdivision in both documents (e.g. *start control line*, *end control line* – Fig. 6; *text line* – Fig. 6 – Note: third subdivision type is the inherent character-by-character type in a line);

comparing current first-type subdivision of one documents to a counterpart of other document; and indicating differences therebetween (e.g. *output table* , *branch deltas* – Fig 7a-b);

comparing current second-type subdivision of one documents to a counterpart of the other document (Fig. 6 – Note: the comparing of lines by lines text is implicitly disclosed);

comparing current third-type subdivision of one documents to a counterpart of the other document (Fig. 6 – Note: the inherent comparing of character-by-character type in a text line is implicitly disclosed);

repeating third comparing step within subdivision of second type (e.g. character in *text lines* - Fig. 6);

repeating second comparing step within subdivision of first type (e.g. *text lines* between *control lines* -- Fig. 6);

repeating first comparing step for further first-type subdivision within the documents(e.g. *control lines* per block - Fig. 6; col. 7, line 30 to col. 8, line 25);

producing an output document indicating differences found in all comparing steps (e.g. *output delta table* - col. 8, lines 16-25; Fig. 5).

As per claim 9, see claim 4 for corresponding rejection.

As per claim 10, Skinner discloses a multi-line section enclosed by control lines (e.g. Fig. 6).

As per claim 11, Skinner discloses a method useful in developing software having multiple versioned documents, comprising: comparing 2 child documents of a common parent document with each other and indicating any differences as actual conflicts between the two (e.g. col. 8, line 55 to col. 9, line 5); comparing both child documents with a common parent for indicating possible conflicts between child documents for portions therein that are same to each other (e.g. *assign level identification, factoring into consideration, incremental deltas* – col. 9, line12-52; Fig. 7a-b – Note: taking into consideration a branch delta as a result of multiple historical version branching by the root document based on level identification is equivalent to

tracking possible differences between apparently identical siblings coming from a parent file); producing a merged output document indicating both actual and possible conflicts (e.g col. 8, 14-27; *combined delta structure file* - Fig. 5).

As per claim 12, see Skinner, Fig. 7a-b (re claim 11).

As per claim 15, this is a computer-readable medium version of claim 11. As for a medium to support the software product, Skinner (Fig. 3) discloses use of workstations to develop the software building, hence implicitly discloses the use of computer-readable medium.

4. Claims 22-26 are rejected under 35 U.S.C. 102(b) as being anticipated by Carrier III et al., USPN: 5,903,897 (hereinafter Carrier).

As per claim 22, Carrier discloses a method useful in developing software having multiple versioned documents, comprising:

associating like versions of versioned documents with each other in a change set specification (e.g. list of *released forms* – col. 6, lines 17-52; Fig. 5-6);

associating additional, non-versioned documents into the same change set (e.g. step 328 – Fig. 9; *build log 550, test cases 520, build report 552*– Fig. 12);

retrieving both versioned and nonversioned documents as a single unit (e.g. *source files and check-in info, defect tracker* -Fig. 11; Fig 12).

As per claim 23, Carrier discloses listing of versioned documents in an association file (e.g. *release forms* → *build 170* – Fig. 3).

As per claim 24, Carrier discloses associating nonversioned and versioned documents by listing them in a same association file (e.g. *build report 552* – Fig. 12).

As per claim 25, Carrier discloses a build generated separately from the list of approved released forms (e.g. build 170 - Fig. 3); hence discloses association file separately stored from versioned documents stored in database *403* (Fig. 11).

As per claim 26, this is a computer medium version of claim 22 with the medium limitation being disclosed by Carrier (medium driver *60* – Fig. 1).

5. Claims 27, 29, and 31-38 are rejected under 35 U.S.C. 102(b) as being anticipated by Leblang et al., USPN: 5,649,200 (hereinafter Leblang).

As per claim 27, Leblang discloses an association file (*configuration record 532* – Fig. 12; col. 25, lines 30-42)or record for a set of versioned documents, comprising: a plurality of entries each designating a version of the versioned documents(e.g. *derived object 500*- Fig. 22); at least one entry designating a non-versioned document pertaining to at least one versioned documents(e.g. *script/.../foo.c*, - Fig. 22).

As per claim 29, Leblang discloses audit or bug report stored in configuration record (e.g. col. 30, line 55 to col. 31, line 10).

As per claim 31, Leblang discloses a method useful in developing software having multiple versioned documents, comprising: synchronizing a set of files from a common storage area to a private enlistment area (e.g. Fig. 1, 7, 12, 15; *VOB, view: alpha* – Fig. 21 – Note: VOB is equivalent to common storage area whereas view is enlistment area); adding a set of build-specific changes to the files in the enlistment area; making local changes to the enlisted files (e.g. Fig. 12,16); thereafter removing the changes from the enlisted files and returning enlistment files to the common area (e.g. *reserved checkouts* - Fig. 14a -Note: check-out of files with save of

original in version control is equivalent to return the non-modified version back into the common area).

As per claim 32, Leblang discloses repeating the enlistment process (e.g. Fig. 7, 19-21).

As per claim 33, Leblang discloses an common area for all separate enlistment areas (e.g. col. 6, lines 4-37 – Note: per host workstation, a VOB is a shared area for all enlistments viewed by the user of that workstation, hence common to all separate views).

As per claim 34, see Leblang, Fig. 7, Figs 11-16; re claim 1.

As per claim 35, Leblang discloses merging with a set of previous local changes (e.g. variant A, variant B, deltaA, deltaB - Fig. 10-16 – Note: changes on delta or variant of a check-out copies are equivalent to merging of previous local changes).

As per claim 36, Leblang discloses getting the build-specific changes from a build area (e.g. Fig. 7, 10-12); and merging the build-specific changes into the enlistment files (e.g. col. 26, line 59 to col. 27, line 58; Fig. 19; *view: alpha* – Fig. 22).

As per claim 37, Leblang discloses selecting a particular build from which to get the changes (e.g. col. 1, line 60 to col. 2, line 29; Fig. 23; *build script* – col. 28, line 62 to col. 29, line 48 – Note: a specific build correspond to a script generated by the developer, and this is equivalent to defining a build specific to generating a script therefor).

As per claim 38, this is a computer medium version of claim 22 with the medium limitation being disclosed by Leblang (e.g. workstation *104* – Fig. 2, 7).

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 1-2 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hopwood et al., USPN: 6,223,343 (hereinafter Hopwood), in view of Skinner, USPN: 5,481,722 (hereinafter Skinner).

As per claim 1, Hopwood discloses a method for developing software having multiple versioned documents, comprising:

comparing multiple ones of the versioned documents at different subdivision level (e.g. *merge elements 16* – Fig. 12; Fig. 15-20; Fig. 8, 11; col. 14, lines 32-36; col. 18, lines 23-46; *merge* -col. 19, lines 36-47 – Note: management of change to level of logical group elements to record changes using merge tool is equivalent to comparing versioned) and indicating the changes on an output document at each subdivision level (e.g. *record of ... changes* - col. 19, line 47 to col. 20, line 10; Fig. 15-20 – Note: tracking changes to different workgroups is equivalent to recording changes to an output document at several level of division);

unmerging from a later version of versioned documents of changes previous to a further set of changes (e.g. *regression, restore, revert* – col. 28, lines 45-55);

associating with the set of changes in versioned documents a plurality of non-versioned documents pertaining respectively to versions of versioned documents (e.g. *issuance control data, maintenance libraries, audit trails, issuance reports, metadata* - col. 13, line 63 to col. 14, line 14; *difference reports* - col. 19, lines 36-40 – Note: each element retrieved for work group build is equivalent to versioned document); updating by copying files from a common storage area to a private enlistment area, adding build-specific changes to enlistment copies, making

changes to the modified copies (e.g. Fig. 3; *host developer workstation* - col. 17, line 50 to col. 18, line 39; *workstation 96* – Fig. 7) and thereafter removing the changes and returning the files to the common area (e.g. *master copy* - col. 18, lines 40-52 – Note: master copy is equivalent to storing/keeping of copies without added changes back into the repository).

But Hopwood does not specify comparing of versioned documents to a common parent document and indicating conflicts caused by alternative histories from the parent document. Hopwood teaches merging and validating of software element with interaction with a repository (e.g. col. 15, lines 14-26; Fig. 8) which suggests a level of conflict resolving with changes incurred life cycle. Skinner, in a method to compare/merge files in a hierarchy of development environments similar to built environment workgroups levels as taught by Hopwood, discloses comparing multiple versioned child documents or deltas to a parent document and recording conflicts (e.g. *input and output tables* - Fig. 7a-b; Fig.8 – Note: *branch deltas* are equivalent to identifying conflicts between offspring versions and ancestor versions via alternative histories). It would have been obvious for one of ordinary skill in the art at the time the invention was made to add to Hopwood's method of merging software elements and auditing changes to child-parent comparison technique as taught by Skinner, in case Hopwood's merge method does not already include one, because this would enable the synchronizing child version from its ancestor version with incremental difference extracting and identification of conflicts by means of tree-like branching, thus enabling better reconciliation of hierarchical versioned documents.

As per claim 2, this is a computer-readable medium version of claim 1. As for a medium to support the software product, Hopwood discloses use of workstations to develop the software building (e.g. Fig. 7) hence implicitly discloses the use of computer-readable medium.

8. Claims 16-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Skinner, USPN: 5,481,722, in view of Hopwood et al., USPN: 6,223,343.

As per claim 16, Skinner discloses a method useful in developing software having multiple versioned documents, comprising:

receiving first, second, and third version levels, where the third version incorporates 2nd set of changes from a second version, and the second version incorporates 1st set of changes from the first version level (e.g. Fig. 8 – Note: C2 include revision levels of P and P inherit levels of GP);

outputting of merged documents (e.g. Fig. 5).

But Skinner does not disclose unmerging the 1st set of changes from a third version level, while preserving the 2nd set of changes. Given the tree lay-out of changes level associated with first, second, third level of versioned documents by Skinner(Fig. 8), it is noted that the possibility for reconciling or removing of set of changes as incorporated in each level as shown is implicitly suggested. Further, Hopwood, in a method to manage elements in workgroup for building a project with multi-level versioned objects (Fig. 11-20) using the merge method analogous to that of Skinner, disclose the possibility to unmerge a version to restore a previous version (e.g. e.g. *regression, restore, revert* – col. 28, lines 45-55). It would have been obvious for one of ordinary skill in the art at the time the invention was made to provide the unmerging of selected set of changes/versioned levels as taught by Hopwood and apply it to the child/parent lay-out suggested by Skinner, because the fact of reverting or retrogress to an earlier set of changes independently of the level in the hierarchy is very helpful in promoting earlier proven versions

that would be fit for release and un-promoting versions unfit for release or that appear to have bugs and needed further re-adjustment.

As per claim 17, this claim is another obvious variation of claim 16 unmerge limitation, hence would be rejected here with the same rationale as set forth therein because once the unmerge is applied to one lower level of the tree, it can be applied higher in the tree or lower in the tree.

As per claim 18, Skinner discloses a developer receiving versioned objects for modification and revision checking hence has implicitly disclosed receiving indications of 1st, 2nd, 3rd version levels (e.g. col. 2, lines 31 to col. 3, line 13; *build table, build list* - Fig. 5).

As per claim 19, this is a computer-readable medium version of claim 16. As for a medium to support the software product, Skinner (Fig. 3) discloses use of workstations to develop the software building, hence implicitly discloses the use of computer-readable medium.

As per claims 20 and 21, these claims correspond to or are slight variations of the selective unmerge limitation of claim 16, hence are rejected herein using the same rationale as set forth therein.

9. Claims 13-14 are rejected under 35 USC 103(a) as being unpatentable over Skinner, USPN: 5,481,722, as applied to claim 11, in view of Howard, USPN: 5,600,834 (hereinafter Howard).

As per claims 13 and 14, Skinner discloses using array structures to store matching input data between file to compare in the merge and reconcile technique (Fig. 5) with indication a possible or conflicts due to alternative histories of child documents but fails to disclose marking of possible conflicts (re claim 13) or actual conflicts (re claim 14) in the merged document.

Howard, in a method to reconcile versions of a file with generation of a merge document analogous to the combined delta structure file by Skinner, disclose a document combining the results from reconciling documents with history of more than one versions (e.g. Fig. 4; col.6, lines 35-48) and marking in this document of possible conflicts due to histories of child/parent documents (e.g. col. 12, lines 11-12). It would have been obvious for one of ordinary skill in the art at the time the invention was made to provide such information marking as suggested by Howard within the combined delta file as taught by Skinner because it would provide additional information for the developer or version administrator to further effect verification or error-proofing on the combined merge output file prior to committing it to persistent storage or distribution to sites as suggested by Howard.

10. Claims 28, 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Leblang et al., USPN: 5,649,200.

As per claim 28, Leblang does not specify including in the association file a design documentation; but Leblang discloses non-versioned documents (e.g. config spec *230* – Fig. 24; build script, header file, operating system in *532* – Fig. 12) such that incorporating specs, operating system, header files and build/shell script is equivalent to including data for documenting the overall environment designed to execute the compiled object. Official notice is taken that documenting design, test and operating environment in a project configuration system was a well-known concept in the art. Hence, it would have been obvious for one of ordinary skill in the art at the time the invention was made to include design documentation in the association file along with system configuration in order to associate versioned objects as taught

by Leblang with the specified design settings under which such objects has been developed to become an executable and deliverable element in the configured build.

As per claim 30, Leblang does not disclose including screen shots in the non-versioned documents although Leblang discloses screen views for resolving version conflict or bugs (e.g. col. 9, line 43 to col. 10, line 37). Official notice is taken that the use of screen captures to take snap shots at error display during a debug or for report on a software execution fault was a well-known practice. Hence, Hence, it would have been obvious for one of ordinary skill in the art at the time the invention was made to include in the association file (*configuration record*) as suggested by Leblang any debug information, e.g. including screen shots as taught by known practices because this would provide graphical evidence for the developer to better address conflicts or bugs thereby ensure quality in delivering versioned elements for the build.

Conclusion

11. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Pat No. 6,256,773 to Bowman-Amuah, disclosing development framework with PVCS and screen shots.

U.S. Pat No. 4,912,637 to Sheedy et al., disclosing line-by-line merging and line identifier.

U.S. Pat No. 6,349,407 to Towfiq, disclosing retention of changes in tree and integration into a copy.

U.S. Pat No. 5,806,078 to Hug et al., disclosing version control and graphical tools for debug and merging.

U.S. Pat No. 6,546,545 to Honarvar et al., disclosing version controlled testing and revert to previous versions.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (703)305-7207. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703)305-9662.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

or faxed to:

(703) 746-7239, (for formal communications intended for entry)

or: (703) 746-7240 (for informal or draft communications, please label "PROPOSED" or "DRAFT")

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington. VA. , 22202. 4th Floor(Receptionist).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

VAT
July 14, 2003

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100